

Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/JP2005/017921

International filing date: 21 September 2005 (21.09.2005)

Document type: Certified copy of priority document

Document details: Country/Office: JP
Number: 2004-286042
Filing date: 30 September 2004 (30.09.2004)

Date of receipt at the International Bureau: 28 October 2005 (28.10.2005)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)



World Intellectual Property Organization (WIPO) - Geneva, Switzerland
Organisation Mondiale de la Propriété Intellectuelle (OMPI) - Genève, Suisse

日本国特許庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: 2004年 9月30日

出願番号
Application Number: 特願2004-286042

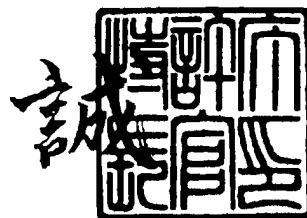
パリ条約による外国への出願
に用いる優先権の主張の基礎
となる出願の国コードと出願
番号
The country code and number
of your priority application,
to be used for filing abroad
under the Paris Convention, is

出願人
Applicant(s): 株式会社リコー

2005年10月12日

特許庁長官
Commissioner,
Japan Patent Office

中嶋



【書類名】 特許願
【整理番号】 200402839
【提出日】 平成16年 9月 30日
【あて先】 特許庁長官殿
【国際特許分類】 G06F 17/30
【発明者】
【住所又は居所】 東京都大田区中馬込1丁目3番6号 株式会社リコー内
【氏名】 山田 孝光
【特許出願人】
【識別番号】 000006747
【氏名又は名称】 株式会社リコー
【代理人】
【識別番号】 100102587
【弁理士】
【氏名又は名称】 渡邊 昌幸
【電話番号】 03-5338-7061
【選任した代理人】
【識別番号】 100077274
【弁理士】
【氏名又は名称】 磐村 雅俊
【電話番号】 03-5338-7061
【連絡先】 担当
【手数料の表示】
【予納台帳番号】 068262
【納付金額】 16,000円
【提出物件の目録】
【物件名】 特許請求の範囲
【物件名】 明細書 1
【物件名】 図面 1
【物件名】 要約書 1
【包括委任状番号】 9808799

【書類名】特許請求の範囲

【請求項1】

半導体集積回路のアサーション検証に用いるアサーション記述を生成するアサーション生成システムであって、

ユーザ操作に基づき上記半導体集積回路の仕様をグラフィカルに編集して当該半導体集積回路の設計データもしくは仕様書及び仕様書確認用のドキュメントを生成する仕様入力手段と、

該仕様入力手段で生成した設計データを記憶する第1の記憶手段と、

該第1の記憶手段から上記仕様入力手段が生成した設計データを読み出し、該設計データを元に、上記半導体集積回路の仕様に関して検証すべきプロパティを生成するプロパティ生成手段と、

該プロパティ生成手段で生成したプロパティを記憶する第2の記憶手段と、

該第2の記憶手段から上記プロパティ生成手段が生成したプロパティを読み出し、該プロパティをアサーション記述に変換するアサーション生成手段と
を有することを特徴とするアサーション生成システム。

【請求項2】

請求項1に記載のアサーション生成システムであって、

上記仕様入力手段は、

ユーザ操作に基づき上記半導体集積回路の仕様を状態遷移表もしくは状態遷移図で編集して当該半導体集積回路の設計データを生成する手段を有することを特徴とするアサーション生成システム。

【請求項3】

請求項1に記載のアサーション生成システムであって、

上記仕様入力手段は、

ユーザ操作に基づき上記半導体集積回路の処理シーケンスをタイミングチャートもしくは時系列に図示して編集し、当該半導体集積回路の設計データを生成する手段を有することを特徴とするアサーション生成システム。

【請求項4】

請求項1から請求項3のいずれかに記載のアサーション生成システムであって、

上記仕様入力手段は、

ユーザ操作に基づき上記半導体集積回路の仕様を論理テーブルもしくは状態テーブルで編集して、当該半導体集積回路の設計データを生成する手段を有することを特徴とするアサーション生成システム。

【請求項5】

請求項4に記載のアサーション生成システムであって、

上記アサーション手段は、

上記仕様入力手段で編集した論理テーブルもしくは状態テーブルのテーブル名あるいはテーブルの行番号、または、当該論理テーブルもしくは状態テーブルにおける信号名あるいは状態名で構成されたアサーション名を付加されたアサーション記述に変換することを特徴とするアサーション生成システム。

【請求項6】

請求項1から請求項5のいずれかに記載のアサーション生成システムであって、

上記仕様入力手段は、

上記設計データをグラフ構造に展開し、上記第1の記憶手段に出力することを特徴とするアサーション生成システム。

【請求項7】

コンピュータを、請求項1から請求項6のいずれかに記載のアサーション生成システムにおける各手段として機能させるためのプログラム。

【請求項8】

請求項1から請求項6のいずれかに記載のアサーション生成システムを具備し、該アサ

ション生成システムで生成したアサーション記述を用いて半導体集積回路のアサーション検証を行うことを特徴とする回路検証システム。

【請求項 9】

半導体集積回路のアサーション検証に用いるアサーション記述を、プログラムされたコンピュータによって生成するアサーション生成方法であって、

ユーザ操作に基づき上記半導体集積回路の仕様をグラフィカルに編集して当該半導体集積回路の設計データを生成して記憶装置に入力する仕様入力手順と、

上記記憶装置から上記仕様入力手順で生成した設計データを読み出し、該設計データを元に、上記半導体集積回路の仕様に関して検証すべきプロパティを生成して記憶装置に入力するプロパティ生成手順と、

上記記憶装置から上記プロパティ生成手順で生成したプロパティを読み出し、該プロパティをアサーション記述に変換するアサーション生成手順と

を有することを特徴とするアサーション生成方法。

【請求項 10】

請求項 9 に記載のアサーション生成方法であって、

上記仕様入力手順では、

ユーザ操作に基づき上記半導体集積回路の仕様を状態遷移表もしくは状態遷移図で編集して当該半導体集積回路の設計データを生成することを特徴とするアサーション生成方法。

【請求項 11】

請求項 9 に記載のアサーション生成方法であって、

上記仕様入力手順では、

ユーザ操作に基づき上記半導体集積回路の処理シーケンスをタイミングチャートもしくは時系列に図示して編集し、当該半導体集積回路の設計データを生成することを特徴とするアサーション生成方法。

【請求項 12】

請求項 9 から請求項 11 のいずれかに記載のアサーション生成方法であって、

上記仕様入力手順では、

ユーザ操作に基づき上記半導体集積回路の仕様を論理テーブルもしくは状態テーブルで編集して、当該半導体集積回路の設計データを生成することを特徴とするアサーション生成方法。

【請求項 13】

請求項 12 に記載のアサーション生成方法であって、

上記アサーション手順では、

上記仕様入力手順で編集した論理テーブルもしくは状態テーブルのテーブル名あるいはテーブルの行番号、または、当該論理テーブルもしくは状態テーブルにおける信号名あるいは状態名で構成されたアサーション名を付加されたアサーション記述に変換することを特徴とするアサーション生成方法。

【請求項 14】

請求項 9 から請求項 13 のいずれかに記載のアサーション生成方法であって、

上記仕様入力手順では、

上記設計データをグラフ構造に展開して上記記憶装置に記憶することを特徴とするアサーション生成方法。

【書類名】明細書

【発明の名称】アサーション生成システムと回路検証システムおよびプログラムならびにアサーション生成方法

【技術分野】

【0001】

本発明は、コンピュータのプログラムに基づく処理により、LSI (Large Scale Integration、大規模集積回路) 等の回路検証を行う技術に係わり、特に、レジスタ転送レベルの設計データをシミュレータでダイナミックに検証するときに、検証対象回路のプロパティをアサーション記述してシミュレータに加味させることで設計違反や検証漏れを確認するようにした検証技術に関するものである。

【0002】

ここで言うプロパティとは、検証対象の設計品に意図・期待される動作を定義する平叙文 (プレーン・テキスト) であり、例えば、アービタ回路における要求信号と承認信号の受信や応答の関係を時系列制約で定義したり、時系列制約の正当状態や、禁止される回路仕様を定義したりするものである。その他、特定の回路シーケンスがシミュレーションでテストされたかどうかを確認する機能カバレッジ・ポイントとして監視すべきイベントとされる。設計記述 (デザイン) が所定のプロパティを満たしているかどうかを検証することをプロパティ検証という。

【背景技術】

【0003】

近年、半導体集積回路においては回路規模の増大化と複雑さのためにコーナーケースの検証漏れや、グループワークのために複数の設計者へ設計担当を割り振った際の各担当ブロック間のインターフェース仕様の確認漏れ、あるいは、第三者から購入した再利用可能なコアの誤仕様などが原因でシステムの改定や改版がしばしば発生している。

【0004】

これらの問題の要因は、十分に回路検証が出来ていなかったか、もしくはコーナーケースや第三者のコアの仕様を十分に理解できていないためにバグを抽出し得るテストシナリオを作成できなかった、などがある。

【0005】

これらのことと背景に、検証用に回路仕様に関するプロパティをアサーション記述してシミュレータ (コンピュータ) に与え、シミュレーション中のアサーション違反の警告や、特定の回路シーケンスがシミュレーションでテストされたかどうかを示す機能カバレッジ・レポートを提供するようにしたアサーション検証技術が提案され、最近では実設計に適用され始めてきた。

【0006】

アサーションは設計の意図 (これをプロパティと指すこともある) を記した注釈であって、通常はRTL (Register Transfer Level) 中にコメント文で記述する。検証期間中はシミュレータによってその意図が解釈され、例えば検証対象の回路がその意図とは異なる動作へ陥った場合はエラー・ログが生成される。ここでの設計の意図とは、例えば回路入力の仮定や前提条件であるほか、それらの条件が満たされた場合に期待する動作といったものを指す。

【0007】

シミュレータは以下の二つのことを行なう。

- (1) 仮定や前提条件といった特定のイベントが発生したかどうか
- (2) そのときの期待動作が正しく完了するかどうか

【0008】

上記 (1) のチェックから、アサーションは特定の回路機能が検証で確認されたかどうかの判定材料をフィードバックする、よってアサーションを検証対象回路へ網羅的に埋め込んでおけばアサーションの被服率 (機能カバレッジと呼ぶ) が得られ、これをもって検証精度を定量化できる。

【0009】

上記(2)のチェックから、アサーションによって検証デバッグへのフィードバックが得られる。例えば、回路の不具合の影響が外部端子で観測困難であるところへ埋め込まれたアサーションは不具合の状況を補足する。また、いかなるスティミュラス(Stimulus)においても仮定や前提条件などの条件が満されるときはいつでもこの期待動作がチェックされる。よって例えばスティミュラスがランダムな系列で構成されていたとしても期待動作との自己照合、すなわち期待値比較が実施される。また、下位ブロックに埋め込まれたアサーションはチップレベル検証においても同様に有効である。

【0010】

これらの設計の意図は論理合成向けRTLと区別された検証向けのVerilog-HDLで記述できるかも知れないが、ここでは別のアサーション言語について述べる。

【0011】

現在、最も普及しているのが非特許文献¹に記載のPSL(Property Specification Language)である。PSLはAccelleraという標準化規格団体によってIEEEへの寄贈が進行中のアサーション言語であり、事実上の標準言語である。以下、PSLによるアサーション記述の具体例を示す。

【0012】

例えば、メモリ制御に関して次のような設計の意図(プロパティ)を検証で監視したいとする。

- 1) read-nとwrite-nは、同時にLowにはならない
- 2) write-nの立ち下がりで、enable-nはHighである
- 3) read-nの立ち下がりで、enable-nはLowである

【0013】

これらをアサーションとしてシミュレータへ渡すためのPSL記述(アサーション記述)は以下の通りである。

【0014】

```
// psl property memcont1 = never (!write-n && !read-n) @ (posedge clk) ;
// psl assert memcont1 ;
// psl property memcont2 = always (enable-n) @ (negedge write-n) ;
// psl assert memcont2 ;
// psl property memcont3 = always (!enable-n) @ (negedge read-n) ;
// psl assert memcont3 ;
```

【0015】

上記PSL記述(アサーション記述)の内の「psl、property、never、assert、always」は全てPSLの予約語である。全ての行はVerilog-HDLのコメントであることを示す「//」で開始する。尚、「//」に代えて「/* */」でPSLの記述全体を閉じても良い。

【0016】

また、上記PSL記述(アサーション記述)の内の「memcont1,2,3, !write-n&&!read-n, posedge clk, enable-n, negedge write-n, !enable-n, negedge read-n」はユーザ定義の記述であって、「memcont1,2,3」以外は上記1)～3)の意図を定義するための具体的な実装データ名、すなわちRTL中に出現するメモリ制御信号名である。

【0017】

また、memcont1～3はアサーション名であって、シミュレータからフィードバックされるエラー・ログや機能カバレッジの識別に使用される。

【0018】

PSLによるアサーション記述は次の構成要素をとる。

【0019】

```
// psl property <アサーション名> =
  <監視すべきイベント> -> <条件が満たされたときの期待動作>
  @<ストローブ条件>;
```

```
// psl assert <アサーション名>:
```

【0020】

例えは1)のアサーションであれば「memcont」がアサーション名、(!write-n && !read-n)が監視すべきイベントであって、この場合は先にneverが付加されている事により()内の条件が決して真とならないことを監視する(仮にalwaysが付加される場合は()内の条件が真であることがチェックされる)。これに続く@(posedge clk)によって条件の監視がclkの立ち上がりエッジのたびに行われることを定義している。「->」は何時、期待動作が開始されるかによって「|=>」、「|->」へ変化する。なお、上記の例は期待動作を付随していない。

【0021】

ストローブ条件以外の各パートの()内はブーリアンを返すVerilog-HDLの式で記述する。ストローブ条件の()内はVerilog-HDLのalways文で記述するセンシティビティリストの記述形式をとる。

【0022】

PSLは監視すべきイベントと、その条件が満たされたときの期待動作の各パートのそれぞれはシーケンスを、すなわち複数サイクルに渡るイベントの変化や期待動作といったものが定義できる。次の例を見てみる

【0023】

・設計の意図：メモリのクリア処理では、クリア開始条件(m-task == 2'b00)が成立したならば、全ワード(256ワード)の初期化のための書き込み操作が256回だけ継続される。ここで、一回の書き込みはclkの立ち上がりサイクルに同期したwrite-nのLow→High→Highの動作で完了する。

【0024】

・PSL記述

```
// psl sequence WRITE-PULSE = { !write-n; write-n; write-n };  
// psl property CLEAR-MEM-WRITE-N =  
// always { m-task == 2'b00 } |=> { WRITE-PULSE [*256] }  
// @(posedge clk);  
// psl assert CLEAR-MEM-WRITE-N;
```

【0025】

sequenceはwrite-nのLow→High→Highといったシーケンスをマクロ定義している。

ここで注目したいのは「|=>」に続く期待動作のパートである。m-task == 2'b00 であれば、WRITE-PULSE、すなわちwrite-nのLow→High→Highといった3サイクルにまたがるシーケンスが256回分、継続することを期待している。

【0026】

しかし、このようなアサーション検証技術においては、例えば、仕様を定義する時に手入力でアサーション記述言語を書いた場合、アサーション自体にミスが混入してしまう可能性がある。その結果、アサーション検証によって得られたアサーション違反の警告や機能カバレッジのレポートと言うものはシステムの改定や改版を防ぐためのフィードバックとはなれず、さらにはアサーション記述言語のデバッグのための検証期間を要してしまい、結果的に検証効率が悪くなってしまう。

【0027】

すなわち、回路仕様とアサーション記述との整合性が一致していることがアサーション検証技術の大前提であるが、シミュレーションを実施するのに先んじてその整合性を確認する手立てがない。また、複雑な有限状態機械に関しては、人手によって網羅的な状態遷移のシーケンスを記述すること自体が困難である。

【0028】

このようなアサーション検証に関しての従来技術としては、例えば、特許文献1および特許文献2に記載の技術がある。

【0029】

特許文献1においては、状態遷移機械をグラフでモデル化し、アサーションを満たすべき状態遷移経路の探索に関する技術が示されている。また、このグラフの解析によってのみアサーションの正当性を確認する技術が示されている。これにより、回路仕様とアサーション記述との整合性を確認することができる。

【0030】

しかし、この技術では、グラフ化する状態遷移機械は、人手で、基礎となる状態遷移機械のモデルを作成するか、もしくは、レジスタ転送レベルの記述言語で作成された設計データから抽出するようにしている。そのため、整合性一致の対象自体、すなわち、回路仕様であるべき状態遷移機械のモデルへのミスの混入は否めない。

【0031】

また、特許文献2においては、レジスタ転送レベルの設計データからデータ転送の構造を抽出してグラフ構造へ展開し、対象回路に関して検証すべきアサーションを生成する技術が記載されている。しかし、この技術では、回路仕様であるアサーションが設計データから抽出されるので、回路仕様とアサーション記述との整合性一致は保証されない。

【0032】

【特許文献1】特開2000-181933号公報

【特許文献2】特開2000-142918号公報

【非特許文献1】「Property Specification Language Reference Manual Version 1.1 June 9, 2004」、[online]、[平成16年9月14日検索]、インターネット<URL : <http://www.eda.org/vlv/docs/PSL-v1.1.pdf>

【発明の開示】

【発明が解決しようとする課題】

【0033】

解決しようとする問題点は、従来の技術では、回路仕様とアサーション記述との整合性一致は保証されない点である。

【0034】

本発明の目的は、これら従来技術の課題を解決し、例えばLSI等に対するアサーション検証の高信頼化および高効率化を可能とすることである。

【課題を解決するための手段】

【0035】

上記目的を達成するため、本発明では、操作者が回路の仕様書を作成するときに使用した視覚的な状態遷移図やタイミングチャート、および、処理のシーケンス図等をもとにし、検証すべきプロパティを自動的にアサーション記述言語へ変換することを特徴とする。

【発明の効果】

【0036】

本発明によれば、例えば仕様書に添付される状態遷移図の電子データから、検証すべきプロパティのアサーション記述を自動的に生成するので、回路仕様とアサーションとの整合性が一致する。これにより、アサーションのデバッグに費やす作業が削除できる。また、生成したアサーションは、状態遷移の経路を網羅的に探索して生成した機能カバレッジ・ポイントであるので、検証の後、との経路がテストされていないかを機能カバレッジ・レポートからフィードバックでき、コーナーケースのテスト漏れを防ぐことができる。また、例えば仕様書に添付するタイミングチャートや信号の選択条件、あるいは処理のシーケンスを注釈もしくは定義した图形もしくは表形式の電子データから、検証すべきプロパティのアサーション記述を自動的に生成するので、回路仕様とアサーションとの整合性が一致する。これにより、アサーションのデバッグに費やす作業が削除できる。特に、仕様書に添付するタイミングチャートや信号の選択条件、または処理のシーケンスというものは検証時に必ずテストされなければならないほど重要なアサーションであることが多く、生成したアサーションは重要プロパティであり、仕様書や回路機能を熟知しない設計者がキー・プロパティを使って検証作業を行うことができ、最終的に検証漏れによるシステムの

改版率を低減させることが可能である。

【発明を実施するための最良の形態】

【0037】

以下、図を用いて本発明を実施するための最良の形態例を説明する。

【0038】

図1は、本発明に係わるアサーション生成システムおよび回路検証システムを構成するハードウェアの構成例を示すブロック図であり、図2は、本発明に係わるアサーション生成システムの第1の機能構成例およびそれを用いた回路検証システムの構成例を示すブロック図、図3は、本発明に係わるアサーション生成システムで用いられるグラフ構造のメモリイメージ例を示す説明図、図4は、本発明に係わるアサーション生成システムで用いられる状態遷移図とそのグラフ構造例を示す説明図、図5は、本発明に係わるアサーション生成システムの第2の機能構成例を示すブロック図、図6は、本発明に係わるアサーション生成システムの第3の機能構成例を示すブロック図、図7は、本発明に係わるアサーション生成システムで用いられるタイミングチャート例を示す説明図、図8は、本発明に係わるアサーション生成システムで用いられるタイミングチャート上で定義される信号に関する情報および当該信号に対するタイミング制約に関する情報を格納するための構造体例を示す説明図、図9は、図7におけるタイミングチャートに関するプロパティ情報のメモリイメージ例を示す説明図、図10は、本発明に係わるアサーション生成システムの第4の機能構成例を示すブロック図、図11は、本発明に係わるアサーション生成システムで用いられるフルアダーテーブルの構成例を示す説明図、図12は、表(テーブル)におけるセルをメモリ上に展開するときの1個のセルを表すための構造体例を示す説明図、図13は、図12におけるセルを表す構造体を組み合わせてシートを構成したときのイメージ例を示す説明図、図14は、図11におけるフルアダーテーブルを図13のメモリ構造へ展開したときのイメージ例を示す説明図、図15は、本発明に係わるアサーション生成システムで用いられる処理シーケンスの表形式での構造例を示す説明図である。

【0039】

図1において、101はメモリもしくはハードディスク等からなる記憶装置、102はCPU(Central Processing Unit、中央処理装置)、103はキーボードやマウスなどからなる入力装置、104はCRT(Cathode Ray Tube)やLCD(Liquid Crystal Display)等からなる表示装置、105はシステムバスであり、システムバス105で各装置101～104を接続することでコンピュータシステムを構成している。

【0040】

記憶装置101では、アーキテクチャに依存しないハードウェア機能記述や本発明に係わる各機能・手段を、CD-ROM(Compact Disc-Read Only Memory)やDVD(Digital Video Disc/Digital Versatile Disc)あるいはFD(Flexible Disk)等のコンピュータ読み取り可能な記録媒体から読み取られた実行可能なプログラムの形式で格納する。CPU102は、記憶装置101に記憶されたプログラムを実行することで、図2に示す構成のアサーション生成システムにおける各機能としての処理を実行する。

【0041】

入力装置103を用いて、ユーザは、アーキテクチャに依存しないハードウェア機能記述の入力・編集や本発明の各手段を実現したプログラムの実行命令を対話的に入力する。そして、表示装置104において、アーキテクチャに依存しないハードウェア機能記述の表示や、CPU102による本発明の各手段を実現したプログラムの実行経過や実行結果を表示する。

【0042】

図2においては、本発明に係るアサーション生成システム207およびそれを用いた回路検証システムの基本構成を示しており、まず、グラフィカルエディタ201において、ユーザ操作により、仕様書に明記すべき回路仕様が入力され、このグラフィカルエディタ201で作成された仕様書は、電子データ202として、記憶装置101に記憶される。

尚、グラフィカルエディタ201としては、例えば、特開平1-309185号公報に記載のタイミングチャート・エディタ等のグラフィカルエディタを用いることができる。

【0043】

次に、構文解析器203により、記憶装置101に記憶された電子データ202を読み出して構文解析し、この構文解析結果から、プロパティ抽出器204により、検証すべきプロパティを抽出する。

【0044】

そして、アサーション生成器205において、プロパティ抽出器204で抽出したプロパティから、シミュレータや静的プロパティチェックへ入力可能なアサーション記述言語206を生成する。このようにして生成されたアサーション記述言語206は、記憶装置101に格納され、シミュレータに入力され、検証が行われる。

【0045】

このように、本例のアサーション生成システムでは、半導体集積回路や半導体集積回路の一部分の仕様に関するプロパティをアサーション記述した設計データと当該プロパティの対象回路の設計データとをシミュレータ、もしくは静的検証ツールで検証するアサーション検証を行うに際に、グラフィカルエディタ201において、有限状態機械の仕様を、状態遷移表や状態遷移図による視覚的表現で編集して仕様書電子データ（設計データ）を生成し、このようにグラフィカルエディタ201において定義した回路仕様に沿し、構文解析器203およびプロパティ抽出器204において、当該設計データを基に検証すべきプロパティを生成し、アサーション生成器205において、このプロパティをアサーション記述によって出力する。

【0046】

尚、グラフィカルエディタ201で作成された設計データ（仕様書電子データ）はメモリ上でグラフ構造に展開され、プロパティ生成抽出器204によって参照可能となっている。また、グラフィカルエディタ201で作成される設計データ（仕様書電子データ）には、状態遷移表や状態遷移図の他に、表入力や図形を編集可能なビジネスツールを用いて回路の処理シーケンスをタイミングチャートや時系列に図示したものも適用できる。

【0047】

図5においては、グラフィカルエディタ201で作成される設計データ（仕様書電子データ）として「状態遷移図」を適用する際の例を示している。本図5において、グラフィカル情報501は、例えば図4（a）に例示される状態遷移図であり、グラフィカルエディタによって編集、作成され、システムメモリ、もしくはハードディスク等の記憶装置に格納されている。

【0048】

また、このグラフィカル情報501は、視覚的情報、すなわち状態遷移図の各構成要素を配置したときの座標情報などを含んでおり、視覚的に表示された状態遷移図はプリントアウトして仕様書に添付できるようになっている。

【0049】

グラフ探索エンジン502は、図2における構文解析器203とプロパティ抽出器204の機能を有し、グラフィカル情報501としての図4（a）に例示する状態遷移図から、図4（b）に例示するグラフ構造を抽出し、システムメモリ上に展開する。抽出したグラフは状態遷移図の各構成要素をノードとして構成されており、グラフ探索エンジン502は、状態遷移の経路を探索する。例えば、状態遷移機械の初期状態からのあらゆる遷移バスを探索し、それらを抽出し、経路データベース（図中「経路BD」と記載）503として、システムメモリに展開、もしくはハードディスク上に格納する。

【0050】

そして、プロパティ変換器504は、経路データベース503を参照して、グラフ探索エンジン502で探索した状態遷移バスをアサーション記述言語505へ変換する。このプロパティ変換器504での変換後のアサーション記述言語505は、例えば、シミュレーションでテストされたかどうかを確認するための機能カバレッジ・ポイントとして記述

される。

【0051】

グラフ探索エンジン502で使用するグラフ構造のメモリイメージを、図3に示す。この図3において、301は状態遷移図中の各状態に関する情報を保持するための構造体であり、302は状態遷移図中のアーチに関する情報を保持するための構造体である。これらの構造体301、302は、相互にポインタ参照しており、それぞれのポインタをノードとしたグラフを構成する。

【0052】

図4においては、図4(a)に状態遷移図の例を示し、それから生成されるグラフ構造を図4(B)に示す。図5のグラフ探索エンジン502の処理は、図4(a)に示す状態遷移図に対して、グラフ理論における経路探索、例えば最短経路探索のアルゴリズムなどをを利用して状態遷移経路を探索するようにしている。その結果として、図4(b)に示すグラフ構造が生成され、経路データベース503に格納される。

【0053】

以下に、図4の例を用いて、図5の経路データベース503に格納される状態遷移バスを例示す。

【0054】

```
Init;  
(G0) S1;  
Init;
```

【0055】

この例では、initからS1への経路が探索されており、しかもそれは「0」が真、すなわち「1」である場合に、その状態遷移バスが成立することを表している。

【0056】

また、この状態遷移バスから、図5のプロパティ変換器504(図2におけるアサーション生成器205)により生成されるアサーション記述をPSLで以下に例示する。

【0057】

```
property init-to-S1 = always { state == init & G0 } => { state == S1 };
```

【0058】

状態遷移機械の場合、検証で全ての状態を訪問したか、と言ったことも回路機能の鍵となる重要機能の検証精度の定量化のために必要となる。状態遷移機械の各状態が訪問されたかどうかはPSLのcover構文で状態の検証の被服率を確認すればよい。図4の例であれば、グラフのノードに格納された状態名をPSLのcover記述でリスト生成するだけよい。

【0059】

以下にその結果を示す。ここでstateは状態遷移機械の実装時に状態を保持するための状態レジスタ名である。

【0060】

```
// psl cover ( state == init );  
// psl cover ( state == S1 );
```

【0061】

尚、図5におけるグラフィカル情報501が、座標情報と共に、図3および図4(b)に示すように、グラフ構造に関する情報を併せ持つようにした場合、グラフ探索エンジン502は、直接、グラフィカル情報501内のグラフ構造を参照する。

【0062】

次に、図6を用いて他の実施例を説明する。本図6に示す例では、仕様書に添付すべきタイミングチャートの電子データ601からタイミングチャート情報を抽出し、検証すべき信号群のタイミングに関する相互関係をアサーション記述するようにしている。

【0063】

図6において、602はタイミングチャートの電子データ601を読みとるタイミング

チャート読み取り器、603は検証すべき信号群のタイミングに関する相互関係（タイミングチャート情報）を抽出してタイミングチャートデータベース（DB）604に格納するタイミング・プロパティ抽出器、605はタイミングチャートデータベース（DB）604からタイミングチャート情報を読み出してシミュレータや静的プロパティチェックへ入力可能なアサーション記述言語606を生成するプロパティ変換器である。

【0064】

尚、タイミングチャートの電子データ601は、タイミングチャート・エディタで作成・編集されたものでよく、その場合のタイミングチャート読み取り器602は、タイミングチャート・エディタで作成・編集されたタイミングチャート情報の構文解析器で構成される。

【0065】

タイミング・プロパティ抽出器603で使用するメモリイメージを図8に示す。本図8において、801はタイミングチャート上で定義される信号に関する情報を格納するための構造体であって、802は前記信号に対して相互にタイミング制約に関する情報を格納するための構造体である。

【0066】

タイミングチャート読み取り器602で読み取るタイミングチャートの具体例を図7に示す。本図7に示す例では、clkに同期してreqの立ち上がりから2～5サイクルの間にackが立ち上がることを示している。

【0067】

このように、図7の例では「reqの出力信号が1になった後、clkの2～5クロックの範囲でackの立ち上がりを受信しなければならない」というタイミング制約（プロパティ）が描かれている。

【0068】

そして、図6のタイミング・プロパティ抽出器603は、IN, OUT, CKなどの信号属性や、立ち上がり条件を意味する「pos」、「->」、「->」のような信号群のタイミングに関する相互関係をキーワードにして検証すべきタイミング情報（プロパティ）を抽出する。

【0069】

この結果、図7に示すタイミングチャート・エディタで作成・編集されたタイミングチャート情報は、例えば、以下のリストに編集されハードウェアに格納される。ここでは、1～3行目はピンと、例えばクロック入力であるとか入出の属性を宣言している。4～7行目は波形情報である。このうち6行目は「clkに同期してreqの立ち上がりから2～5サイクルの間にackが立ち上がる」というプロパティが示されている。

【0070】

```
1 CK      clk;
2 IN      ack;
3 OUT     req;
4 {clk, req, ack} = {1, 0, 0};
5 {clk, req, ack} = {1, 1, 0};
6 {clk, req, ack} = {1, 0, 0}[2:5];
7 {clk, req, ack} = {1, 0, 1};
```

【0071】

このタイミングチャート情報リストをもとに展開されたタイミングに関するプロパティ情報のメモリイメージを、図9に例示する。

【0072】

次に、上述した回路の論理テーブルや時系列プロパティの編集に、表入力や図形の編集可能なビジネスツールを用いる例を説明する。

【0073】

図11は、フルアダマーの論理テーブルをマイクロソフト社の表入力ツールであるExcel（登録商標）で入力したものである。太字で示したNAME, CLOCK, CONDITION, EXPECTは

表定義する際の予約語であって、それぞれプロパティ名、クロック名、アサーション・チェックの開始条件、アサーション・チェックが開始されたときの期待動作を表す。

【0074】

また、CONDITION, EXPECTのすぐ下の欄は信号名であり、その下には論理値が続いている。このExcel（登録商標）によれば表の一こまを「セル」という基本単位で面（シートと呼ばれる）を構成しており、各セルを「.」で区切ったASCIIファイルで出力可能である。

【0075】

図7の場合は、信号名やクロックの1/2パルス等をセルで管理し、いったんメモリ上へ展開した後にセルの構成を走査して、図9に示すメモリイメージへ展開するようにした。

【0076】

図12は、Excel（登録商標）のセルをメモリ上に展開するときの1個のセルを表すための構造体である。この構造体のtop, bottom, left, rightのメンバは、それぞれセルの上下、左右をポインタ参照する。

【0077】

図13は、図12のセルを表す構造体を組み合わせてシートを構成したときのイメージ図である。1301はシートを示すポインタであって、次のシートのポインタ参照であるnextとセルのヘッダをポインタ参照するメンバで構成している。1302はセルのポインタであって、top, bottom, left, rightのメンバで上下、左右をポインタ参照している。こうしておけばシート上のセルの走査がメモリ上で容易に可能であり、NAME, CLOCK, CONDITION, EXPECTといったキーワードによる検索と論理テーブルの解釈が可能である。

【0078】

図14には、図11のフルアダマー論理テーブルを図13のメモリ構造へ展開したときのイメージを示している。

【0079】

また、以下に、図11で示したフルアダマー論理テーブルからアサーション記述を生成したPSL記述を示す。ここでは、アサーション名の構成に表の行番号を含ませることで表と検証データ（すなわちアサーション）とが容易にリンクできるようにしている。例えば1行目であればFADD-S-0--FADD-csv-line-5がアサーション名であって、このうちline-5は表の行番号に相当している。

【0080】

```
psl property FADD-S-0--FADD-csv-line-5 = always {!A & !B & !C} |-> {S == 1'b0}
;
psl property FADD-C0-0--FADD-csv-line-5 = always {!A & !B & !C} |-> {C0 == 1'b0};
psl property FADD-S-1--FADD-csv-line-6 = always {!A & !B & C} |-> {S == 1'b1};
psl property FADD-C0-1--FADD-csv-line-6 = always {!A & !B & C} |-> {C0 == 1'b0};
;
psl property FADD-S-2--FADD-csv-line-7 = always {!A & B & !C} |-> {S == 1'b1};
psl property FADD-C0-2--FADD-csv-line-7 = always {!A & B & !C} |-> {C0 == 1'b0};
;
psl property FADD-S-3--FADD-csv-line-8 = always {!A & B & C} |-> {S == 1'b0};
psl property FADD-C0-3--FADD-csv-line-8 = always {!A & B & C} |-> {C0 == 1'b1};
;
psl property FADD-S-4--FADD-csv-line-9 = always {A & !B & !C} |-> {S == 1'b1};
psl property FADD-C0-4--FADD-csv-line-9 = always {A & !B & !C} |-> {C0 == 1'b0};
;
psl property FADD-S-5--FADD-csv-line-10 = always {A & !B & C} |-> {S == 1'b0};
psl property FADD-C0-5--FADD-csv-line-10 = always {A & !B & C} |-> {C0 == 1'b1}
```

```

};

psl property FADD-S-6--FADD-csv-line-11 = always {A & B & !C} |-> {S === 1'b0};
psl property FADD-C0-6--FADD-csv-line-11 = always {A & B & !C} |-> {C0 === 1'b1};
};

psl property FADD-S-7--FADD-csv-line-12 = always {A & B & C} |-> {S === 1'b1};
psl property FADD-C0-7--FADD-csv-line-12 = always {A & B & C} |-> {C0 === 1'b1};
;

```

【0081】

図10において、本発明に係わる他の実施例を示す。この例では仕様書に添付すべきシーケンスを表データ1001として格納しており、構文解析器1002によって、表形式で定義されたシーケンスをシステムメモリ上に展開し、シーケンス抽出器1003によって、検証すべき処理のシーケンスを抽出し、プロパティ変換器1004によって、アサーション記述1005へ変換するようにしている。

【0082】

表形式による処理のシーケンスの例を図15に示す。この例は図7と同じプロパティを示している。この例もExcel（登録商標）で作成されている。構文解析器1002やシーケンス抽出器1003は、セル内に記述されたシーケンスを時系列に分解し、プロパティ変換器1004において、PSLのSERE拡張正規表現でアサーション記述を生成するようにしている。

【0083】

シーケンスの時系列への分解は、文字列内の「;」をクロックの1サイクル間の振る舞いとして解釈する。また、図15に示す「0[*2:5]」はPSLの繰り返し表現を用いており、Low期間が2～5サイクル間だけ継続することを意味している。また、「ack」は「0[*2:5];1;0」と記されているので、「ack」はLow期間が2～5サイクル間だけ継続し、その後、High, Lowと続く振る舞いであることを示している。

【0084】

以下に、図15を元に生成されたPSLのアサーション記述を示す。

【0085】

```

psl property reqack-ack-0--REQ-csv-line-5 =
    always {{(rose)} |-> {{ack === 1'b0}[*2:5];{ack === 1'b1};{ack === 1'b0}}};

```

【0086】

尚、図5と図6および図10に示された各処理フローを、図1におけるCPU102が実行するためのプログラムは、実行可能な形式で磁気メディア等に記録され、図1の記憶装置101に格納される。

【0087】

以上、図1～図15を用いて説明したように、本例のアサーション生成システム207は、半導体集積回路のアサーション検証に用いるアサーション記述を生成するものであって、グラフィカルエディタ（仕様入力手段）201により、ユーザ操作に基づき半導体集積回路の仕様（有限状態機械、処理シーケンス）を、状態遷移表や状態遷移図、あるいは、タイミングチャートや時系列图形で、グラフィカルに編集して当該半導体集積回路の設計データを生成し、グラフ構造に展開して仕様書電子データ202とし、記憶装置に記憶し、構文解析器203およびプロパティ抽出器204（プロパティ生成手段）により、仕様書電子データ（設計データ）を記憶装置から読み出し、この設計データを元に、半導体集積回路の仕様に関して検証すべきプロパティを生成し、そして、アサーション生成器（アサーション生成手段）205により、当該プロパティをアサーション記述言語206に変換する。

【0088】

尚、グラフィカルエディタ（仕様入力手段）201では、ユーザ操作に基づき半導体集積回路の仕様を論理テーブルもしくは状態テーブルで編集して、当該半導体集積回路の設計データを生成することでも良い。そして、アサーション生成器205は、グラフィカル

エディタ（仕様入力手段）201で編集した論理テーブルもしくは状態テーブルのテーブル名あるいはテーブルの行番号、または、当該論理テーブルもしくは状態テーブルにおける信号名あるいは状態名で構成されたアサーション名を付加されたアサーション記述に変換する。

【0089】

また、本例の回路検証システムは、このようなアサーション生成システム207を具備し、このアサーション生成システムで生成したアサーション記述を用いて半導体集積回路のアサーション検証を行う。

【0090】

このように、本例では、仕様書に添付する状態遷移図の電子データから、検証すべきプロパティのアサーション記述を自動的に生成するようにしたので、回路仕様とアサーションとの整合性が一致する。そのためアサーションのデバッグに費やす作業が削除できる。また生成されたアサーションは状態遷移の経路を網羅的に探索されて生成された機能カバレッジ・ポイントであるので、検証の後、どの経路がテストされていないかを機能カバレッジ・レポートからフィードバックでき、コーナー・ケースのテスト漏れを防ぐことが出来る。

【0091】

また、仕様書に添付するタイミングチャートや信号の選択条件、または処理のシーケンスを注釈もしくは定義した图形もしくは表形式の電子データから、検証すべきプロパティのアサーション記述を自動的に生成するようにしたので、回路仕様とアサーションとの整合性が一致する。そのためアサーションのデバッグに費やす作業が削除できる。また仕様書に添付するタイミングチャートや信号の選択条件、または処理のシーケンスというものは検証時に必ずテストされなければならないほど重要なアサーションであることが多いが、本例で生成されるアサーションは重要プロパティであり、仕様書や回路機能を熟知しない設計者がキー・プロパティを使って検証作業を行うことができ、最終的に検証漏れによるシステムの改版率を低減させることができる。

【0092】

尚、本発明は、図1～図15を用いて説明した例に限定されるものではなく、その要旨を逸脱しない範囲において種々変更可能である。例えば、本例では、プロパティ抽出器204やアサーション生成器205等を構成するプログラムは、CD-ROM (Compact Disc-Read Only Memory) やDVD (Digital Video Disc/Digital Versatile Disc) あるいはFD (Flexible Disk) 等の記録媒体から記憶装置101にインストールする構成としているが、このようなプログラムのインストールに関しては、通信装置を介してネットワーク経由でプログラムをダウンロードしてインストールすることでも良い。

【0093】

また、本例では、PSLでのアサーション記述を例に説明したが、他のアサーション記述にも適用可能である。

【図面の簡単な説明】

【0094】

【図1】本発明に係わるアサーション生成システムおよび回路検証システムを構成するハードウェアの構成例を示すブロック図である。

【図2】本発明に係わるアサーション生成システムの第1の機能構成例およびそれを用いた回路検証システムの構成例を示すブロック図である。

【図3】本発明に係わるアサーション生成システムで用いられるグラフ構造のメモリイメージ例を示す説明図である。

【図4】本発明に係わるアサーション生成システムで用いられる状態遷移図とそのグラフ構造例を示す説明図である。

【図5】本発明に係わるアサーション生成システムの第2の機能構成例を示すブロック図である。

【図6】本発明に係わるアサーション生成システムの第3の機能構成例を示すブロック図である。

ク図である。

【図7】本発明に係わるアサーション生成システムで用いられるタイミングチャート例を示す説明図である。

【図8】本発明に係わるアサーション生成システムで用いられるタイミングチャート上で定義される信号に関する情報および当該信号に対するタイミング制約に関する情報を格納するための構造体例を示す説明図である。

【図9】図7におけるタイミングチャートに関するプロパティ情報のメモリイメージ例を示す説明図である。

【図10】本発明に係わるアサーション生成システムの第4の機能構成例を示すプロック図である。

【図11】本発明に係わるアサーション生成システムで用いられるフルアダーテーブルの構成例を示す説明図である。

【図12】表(テーブル)におけるセルをメモリ上に展開するときの1個のセルを表すための構造体例を示す説明図である。

【図13】図12におけるセルを表す構造体を組み合わせてシートを構成したときのイメージ例を示す説明図である。

【図14】図11におけるフルアダーテーブルを図13のメモリ構造へ展開したときのイメージ例を示す説明図である。

【図15】本発明に係わるアサーション生成システムで用いられる処理シーケンスの表形式での構造例を示す説明図である。

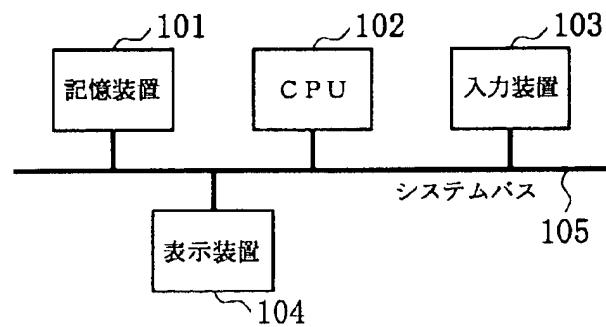
【符号の説明】

【0095】

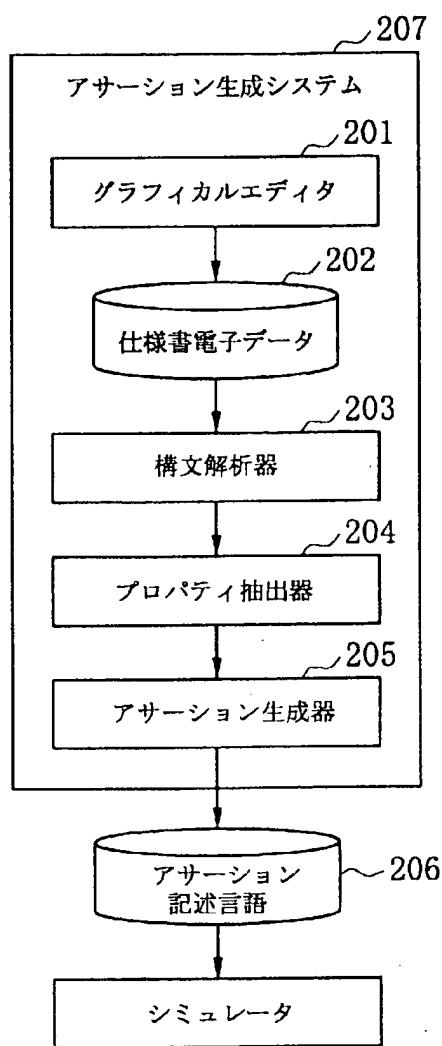
101：記憶装置、102：CPU、103：入力装置、104：表示装置、105：システムバス、201：グラフィカルエディタ、202：電子データ、203：構文解析器、204：プロパティ抽出器、205：アサーション生成器、206：アサーション記述言語、301：構造体(アーケの集合へのポインタ)、302：構造体(Nextアーケ・ポインタ)、501：グラフィカル情報、502：グラフ探索エンジン、503：経路データベース(DB)、504：プロパティ変換器、505：アサーション記述言語、601：タイミングチャート電子データ、602：タイミングチャート読み取り器、603：タイミング・プロパティ抽出器、604：タイミングチャートデータベース(DB)、605：プロパティ変換器、606：アサーション記述言語、801：構造体(関連信号へのポインタ)、802：構造体(Next制約ポインタ)、1001：表データ、1002：構文解析器、1003：シーケンス抽出器、1004：プロパティ変換器、1005：アサーション記述言語、1301：ポインタ(シートを示す)、1302：ポインタ(セル)。

【書類名】 図面

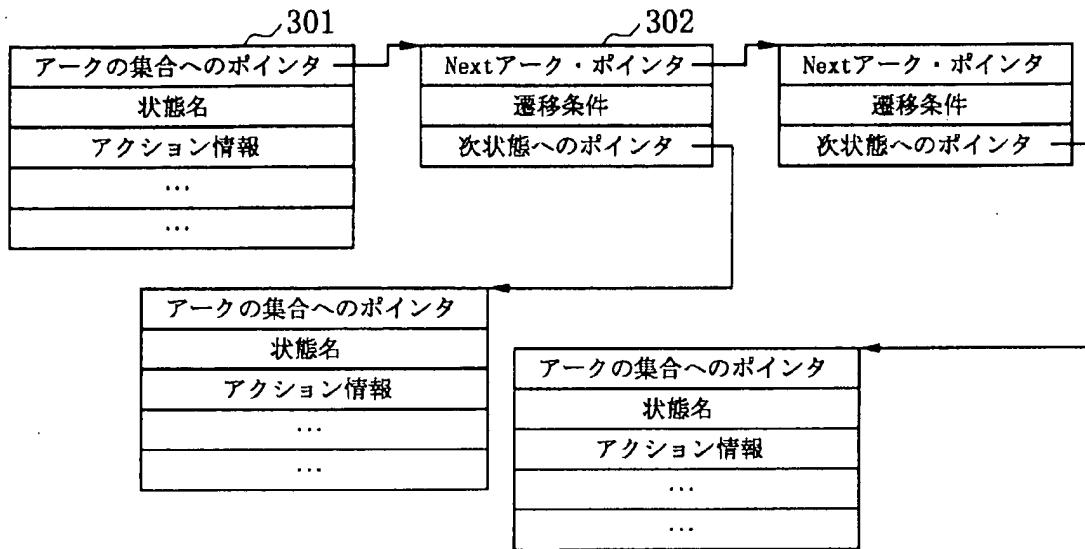
【図 1】



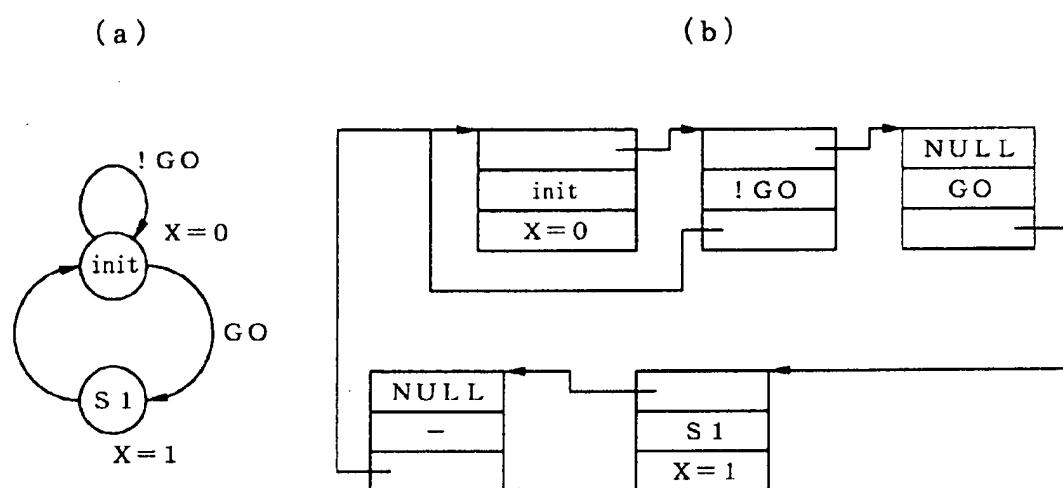
【図2】



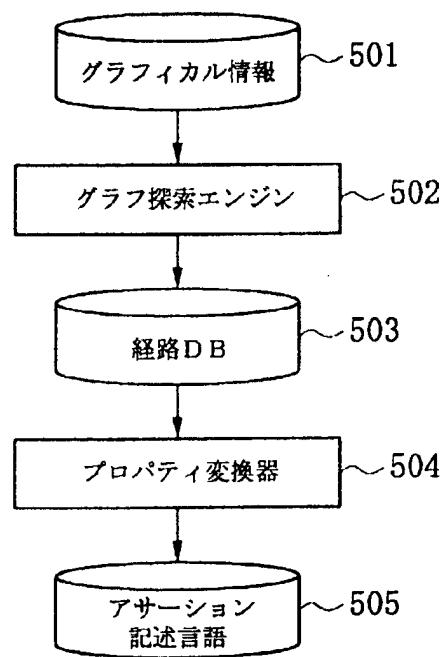
【図 3】



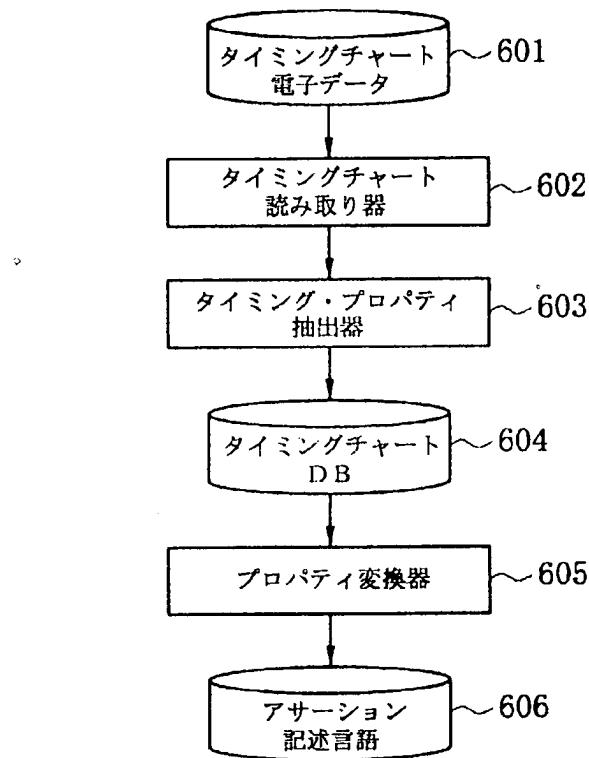
【図 4】



【図 5】



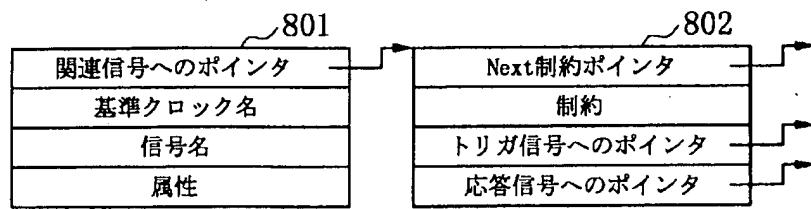
【図 6】



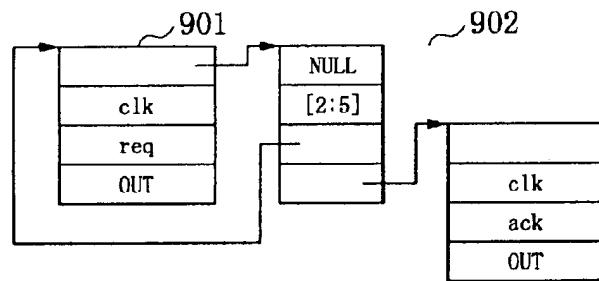
[図 7]

		<- [2:5] ->				
req	(OUT)		pos			
ack	(IN)				pos	
clk	(CK)					

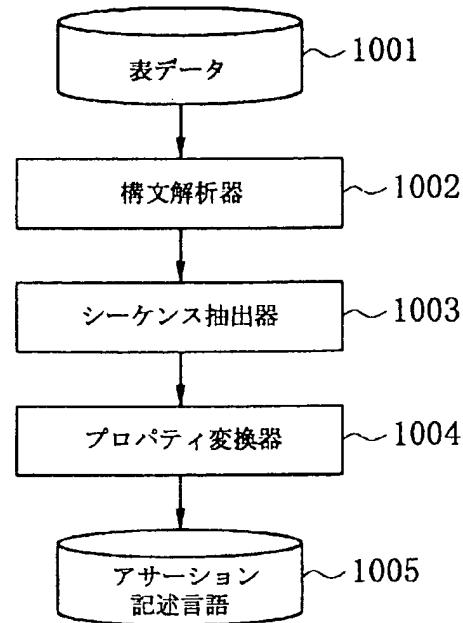
【図 8】



【図 9】



【図 1 0】



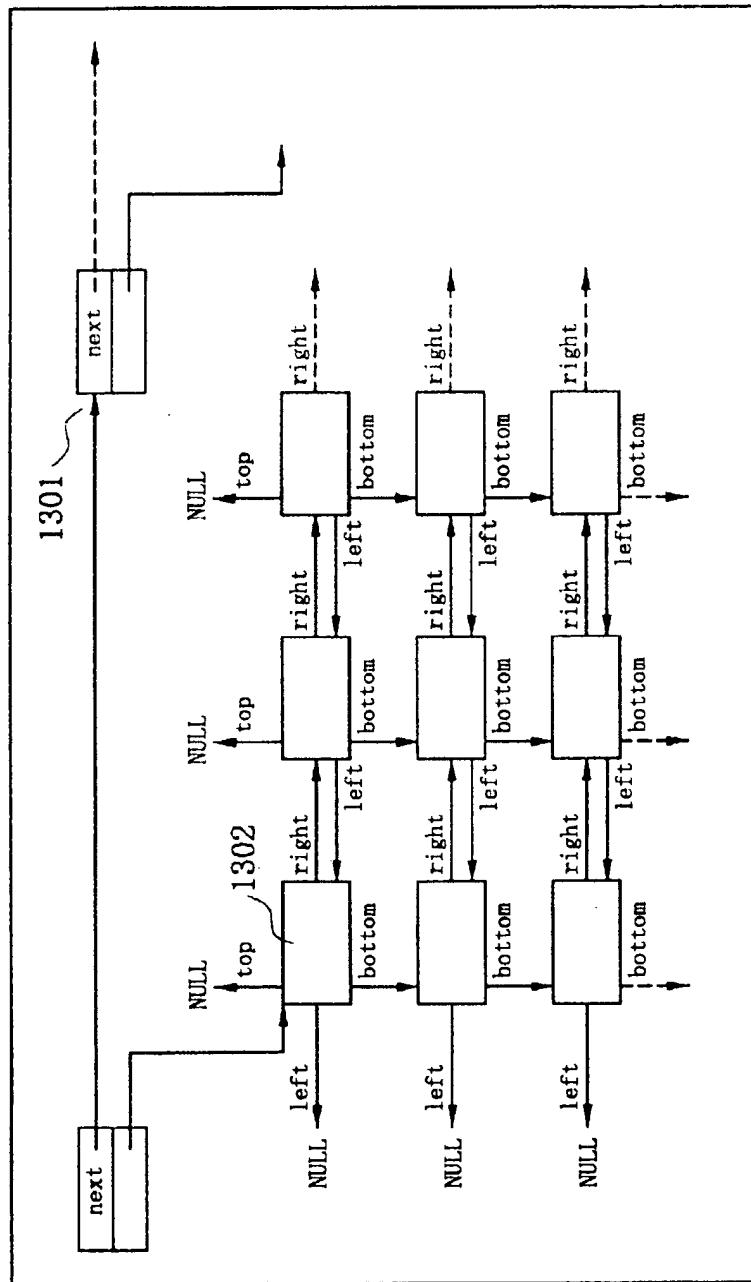
【図 1 1】

<u>NAME</u>	<u>FADD</u>			
<u>CLOCK</u>	<u>posedge clk</u>			
<u>CONDITION</u>	<u>CONDITION</u>	<u>CONDITION</u>	<u>EXPECT</u>	<u>EXPECT</u>
<u>A</u>	<u>B</u>	<u>C1</u>	<u>S</u>	<u>C0</u>
<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>
<u>0</u>	<u>0</u>	<u>1</u>	<u>1</u>	<u>0</u>
<u>0</u>	<u>1</u>	<u>0</u>	<u>1</u>	<u>0</u>
<u>0</u>	<u>1</u>	<u>1</u>	<u>0</u>	<u>1</u>
<u>1</u>	<u>0</u>	<u>0</u>	<u>1</u>	<u>0</u>
<u>1</u>	<u>0</u>	<u>1</u>	<u>0</u>	<u>1</u>
<u>1</u>	<u>1</u>	<u>0</u>	<u>0</u>	<u>1</u>
<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>

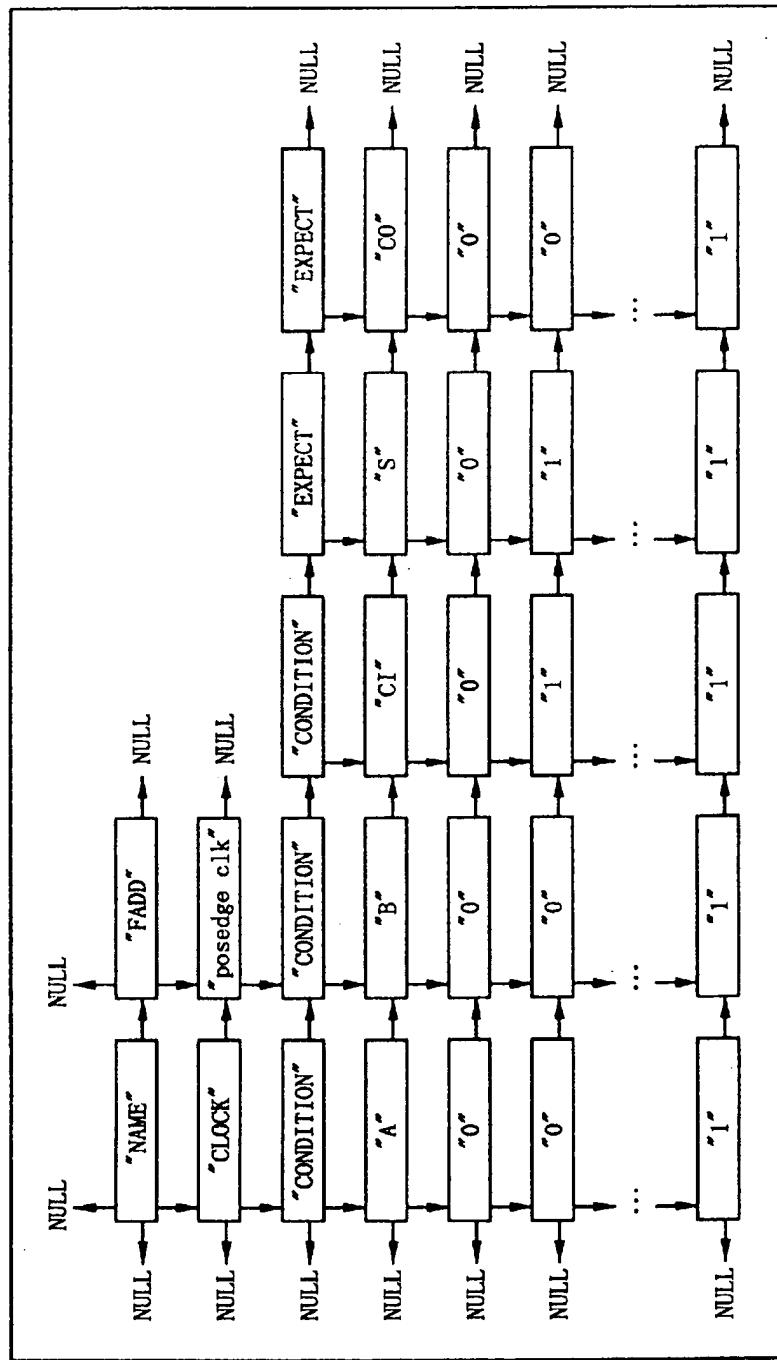
【図 1-2】

```
typedef struct piece {
    struct piece      *left, *right, *top, *bottom
    char*             str;
} Piece;
```

[図 13]



【図 1-4】



<u>NAME</u>	<u>reqack</u>
<u>CLOCK</u>	<u>posedge clk</u>
<u>CONDITION</u>	<u>EXPECT</u>
<u>req</u>	<u>ack</u>
<u>rose</u>	<u>0[*2:5]:1;0</u>

【書類名】要約書

【要約】

【課題】 従来の技術では、回路仕様とアサーション記述との整合性一致は保証されないとの課題を解決し、例えばLSI等に対するアサーション検証の高信頼化および高効率化を可能とする。

【課題を解決するための手段】

【解決手段】 本例のアサーション生成システム207は、グラフィカルエディタ（仕様入力手段）201により、ユーザ操作に基づき半導体集積回路の仕様（有限状態機械、処理シーケンス）を、状態遷移表や状態遷移図あるいはタイミングチャートや時系列図形で、グラフィカルに編集して当該半導体集積回路の設計データ（仕様書電子データ202）を生成し、構文解析器203およびプロパティ抽出器204からなるプロパティ生成手段により、設計データを元に、半導体集積回路の仕様に関して検証すべきプロパティを生成し、アサーション生成器205により、当該プロパティをアサーション記述言語206に変換する。

【選択図】

図2

出願人履歴

000006747

20020517

住所変更

東京都大田区中馬込1丁目3番6号

株式会社リコー